Practical 1

AIM: Implement Encryption-Decryption for below ciphers:

a) Caesar cipher
b) Brute force attack on cipher

Ans:

-----Caesar cipher

```
#include<iostream>
using namespace std;

string encryption(string plain,int key){
	string enc;
	for(int i=0;i<plain.length();i++){
		if(plain[i]==32)
			enc+=plain[i];
		else
			enc+=(plain[i]+key-97)%26+97;
	}
	return enc;
}

string decryption(string enc,int key){
	string plain;
	for(int i=0;i<enc.length();i++){
		if(enc[i]==32)
			plain+=enc[i];
		else{
			char test=(enc[i]-key-97)%26;
			if(test<0)
				test+=123;
			else
				test+=97;

			plain+=test;
		}
	}
	return plain;
}
```

```
int main(){
        int key;
        string plain;
        cout<<"Enter the Plain Text:";
        getline(cin,plain);
        cout<<"Enter the Key:";
        cin>>key;
        cout<<endl<<"Plain Text:"<<plain<<endl;
        cout<<"Encryption Text:"<<encryption(plain,key)<<endl;
        string enc=encryption(plain,key);
        cout<<"Decryption Text:"<<decryption(enc,key);
}
```

Output

D:\P1.exe

```
Enter the Plain Text:dhruvil birenkumar shah
Enter the Key:5

Plain Text:dhruvil birenkumar shah
Encryption Text:imwzanq gnwjspzrfw xmfm
Decryption Text:dhruvil birenkumar shah
--------------------------------
Process exited after 9.031 seconds with return value 0
Press any key to continue . . .
```

------- Brute force attack on cipher

```
#include<iostream>
using namespace std;

string encryption(string plain,int key){
	string enc;
	for(int i=0;i<plain.length();i++){
		if(plain[i]==32)
			enc+=plain[i];
		else
			enc+=(plain[i]+key-97)%26+97;
	}
	return enc;
}

string decryption(string enc,int key){
	string plain;
	for(int i=0;i<enc.length();i++){
		if(enc[i]==32)
			plain+=enc[i];
		else{
			char test=(enc[i]-key-97)%26;
			if(test<0)
				test+=123;
			else
				test+=97;
```

```
                plain+=test;
            }
        }
        return plain;
}

void brute_force(string enc){
        for(int j=1;j<26;j++){
            string plain="";
            for(int i=0;i<enc.length();i++){
                if(enc[i]==32)
                    plain+=enc[i];
                else
                    plain+=(enc[i]+j-97)%26 + 97;
            }
            cout<<"Key "<<j<<" value : "<<plain<<endl;
        }
}

int main(){
        int key;
        string plain;
        cout<<"Enter the Plain Text:";
        getline(cin,plain);
        cout<<"Enter the Key:";
        cin>>key;
```

```
        cout<<endl<<"Plain Text:"<<plain<<endl;
        cout<<"Encryption Text:"<<encryption(plain,key)<<endl;
        string enc=encryption(plain,key);
        cout<<"Decryption Text:"<<decryption(enc,key)<<endl;
        cout<<endl<<"----Brute Force Attack----"<<endl;
        brute_force(enc);
}
```

Output:

D:\P2.exe

```
Enter the Plain Text:dhruvil birenkumar shah
Enter the Key:5

Plain Text:dhruvil birenkumar shah
Encryption Text:imwzanq gnwjspzrfw xmfm
Decryption Text:dhruvil birenkumar shah

----Brute Force Attack----
Key 1 value : jnxabor hoxktqasgx yngn
Key 2 value : koybcps ipylurbthy zoho
Key 3 value : lpzcdqt jqzmvscuiz apip
Key 4 value : mqaderu kranwtdvja bqjq
Key 5 value : nrbefsv lsboxuewkb crkr
Key 6 value : oscfgtw mtcpyvfxlc dsls
Key 7 value : ptdghux nudqzwgymd etmt
Key 8 value : quehivy overaxhzne funu
Key 9 value : rvfijwz pwfsbyiaof gvov
Key 10 value : swgjkxa qxgtczjbpg hwpw
Key 11 value : txhklyb ryhudakcqh ixqx
Key 12 value : uyilmzc szivebldri jyry
Key 13 value : vzjmnad tajwfcmesj kzsz
Key 14 value : waknobe ubkxgdnftk lata
Key 15 value : xblopcf vclyheogul mbub
Key 16 value : ycmpqdg wdmzifphvm ncvc
Key 17 value : zdnqreh xenajgqiwn odwd
Key 18 value : aeorsfi yfobkhrjxo pexe
Key 19 value : bfpstgj zgpcliskyp qfyf
Key 20 value : cgqtuhk ahqdmjtlzq rgzg
Key 21 value : dhruvil birenkumar shah
Key 22 value : eisvwjm cjsfolvnbs tibi
Key 23 value : fjtwxkn dktgpmwoct ujcj
Key 24 value : gkuxylo eluhqnxpdu vkdk
Key 25 value : hlvyzmp fmviroyqev wlel

--------------------------------
Process exited after 10.93 seconds with return value 0
Press any key to continue . . .
```

Practical 2

AIM: Implement Rail-fence cipher encryption-decryption

Ans:

```
#include <bits/stdc++.h>

#include<iostream>

using namespace std;


string encode(string original, int key){

	string ans="";

	int len=original.length();

	vector<string> v(key);

	int k = 0;

	while(k!=len){

		for(int i=0;i<key-1;i++){

			v[i]+=original[k++];

			if(k==len)

				break;

		}

		if (k==len)

			break;

		for(int i=key-1;i>=1;i--){

			v[i]+=original[k++];

			if (k==len)

				break;

		}

	}

	for (int i=0;i<key;i++){
```

```
		ans+=v[i];
	}
	return ans;
}

string decode(string str,int key){
	int len=str.length();
	int rep=len/(2*(key-1));
	int rem=len%(2*(key-1));
	vector<int> space(key,2*rep);
	space[0]=rep;
	space[key-1]=rep;
	if(rem!=0){
		for (int i=0;i<key-1;i++){
			space[i]++;
			rem--;
			if(rem==0)
				break;
		}
	}
	if(rem!=0){
		for (int i=key-1;i>=1;i--){
			space[i]++;
			rem--;
			if (rem==0)
				break;
		}
```

```
        }
        vector<string> v;
        int start=0;
        for (int i=0;i<key;i++){
                v.push_back(str.substr(start,space[i]));
                start += space[i];
        }
        vector<int> place(key, 0);
        string ans="";
        int k=0;
        while(ans.length()!=len){
                for (int i=0;i<key-1;i++){
                        ans+=v[i][place[i]++];
                        if(ans.length()==len)
                                break;
                }
                if(ans.length()==len)
                        break;
                for (int i=key-1;i>=1;i--){
                        ans+=v[i][place[i]++];
                        if(ans.length()==len)
                                break;
                }
        }
        return ans;
}
```

```
int main(){
        int key;
        string original="";
        cout<<"Enter the Message to encrypt:";
        getline(cin, original);
        cout<<"How many lanes you want : ";
        cin>>key;

        string encrypted=encode(original, key);
        cout<<"Encrypted : "<<encrypted<<endl;

        string decrypted = decode(encrypted, key);
        cout<<"Decrypted : "<<decrypted<<endl;
}
```

Output:

C:\Users\dhruv\Downloads\Practical2.exe

```
Enter the Message to encrypt:dhruvil shah
How many lanes you want : 3
Encrypted : dvshui hhrla
Decrypted : dhruvil shah

--------------------------------
Process exited after 16.01 seconds with return value 0
Press any key to continue . . .
```

Practical 3

AIM: Implement Playfair cipher encryption-decryption

Ans:

```
#include<bits/stdc++.h>
#include<iostream>
using namespace std;

vector<string> createTable(string key){
	string ans="";
	bitset<26> present=0;
	for(char x:key){
		if(present[x-'a']==0){
			if(x=='j')
			x='i';
			ans+=x;
			present[x-'a']=1;
		}
	}
	present['j'-'a']=1;
	key=ans;
	int len=key.size();
	vector<string> table(5,"*****");
	int i=0,j=0,k=0;

	while(k!=len){
		table[i][j]=key[k];
```

```
		k++;
		j++;
		if(j==5){
			j=0;
			i++;
		}
	}

	for(int k=0;k<26;k++){
		if(present[k]==0){
			char c=k+'a';
			table[i][j]=c;
			j++;
			if(j==5){
				j=0;
				i++;
			}
		}
	}
	return table;
}

string encrypt(vector<string> table,string input){
	int len=input.size();
	string test="",ans="";
	vector<pair<char,char>> chunk;
	for(auto x:input){
```

```
        if(x<='z' && x>='a'){
                if(x=='j')
                        x='i';
                test+=x;
        }
}
input=test;
len=input.length();
if(len%2==1){
        input+='x';
        len++;
}
cout<<"Trimmed string  : " << input << endl;
for(int i=1;i<len;i+=2)
{
        if(input[i-1]==input[i])
                chunk.push_back(make_pair(input[i-1],'x'));
        else
                chunk.push_back(make_pair(input[i-1],input[i]));
}

unordered_map<char,pair<int,int>> place;
for(int i=0;i<5;i++)
        for(int j=0;j<5;j++)
                place[table[i][j]]=make_pair(i,j);
```

```
	for(auto x:chunk){
		pair<int,int> p = place[x.first];
		pair<int,int> q = place[x.second];
		if(p.first==q.first){
			int f = (p.second+1)%5;
			int s = (q.second+1)%5;
			ans+=table[p.first][f];
			ans+=table[p.first][s];
		}
		else if(p.second==q.second){
			int f = (p.first+1)%5;
			int s = (q.first+1)%5;
			ans+=table[f][p.second];
			ans+=table[s][p.second];
		}
		else{
			ans+=table[p.first][q.second];
			ans+=table[q.first][p.second];
		}
	}
	return ans;
}

string decrypt(vector<string> table,string code){
	string ans="";
	int len=code.length();
	unordered_map<char,pair<int,int>> place;
```

```
	for(int i=0;i<5;i++)
		for(int j=0;j<5;j++)
			place[table[i][j]]=make_pair(i,j);
	for(int i=1;i<len;i+=2){
		pair<int,int> p = place[code[i-1]];
		pair<int,int> q = place[code[i]];
		if(p.first==q.first){
			int f=(p.second+4)%5;
			int s=(q.second+4)%5;
			ans+=table[p.first][f];
			ans+=table[p.first][s];
		}
		else if(p.second==q.second){
			int f=(p.first+4)%5;
			int s=(q.first+4)%5;
			ans+=table[f][p.second];
			ans+=table[s][p.second];
		}
		else{
			ans+=table[p.first][q.second];
			ans+=table[q.first][p.second];
		}
	}
	return ans;
}

int main(){
```

```
    string key="",input="";
    cout<<"Enter the String you want to encode : "; getline(cin,input);
    cout<<"Enter Key for PlayFair cipher : ";cin>>key;
    vector<string> table = createTable(key);
    string encrypted = encrypt(table,input);
    cout<<"Encrypted : "<<encrypted<<endl;
    string decrypted = decrypt(table,encrypted);
    cout<<"Decrypted : "<<decrypted<<endl;
}
```

Output:

```
Enter the String you want to encode : my name is dhruvil shah i am backend developer
Enter Key for PlayFair cipher : vgecisbest
Trimmed string  : mynameisdhruvilshahiambackenddeveloper
Encrypted : lzqskivdbmnzgvfalbmgdltdelvptzcgckpqip
Decrypted : mynameisdhruvilshahiambackendxeveloper

--------------------------------
Process exited after 25.91 seconds with return value 0
Press any key to continue . . .
```